# ネットワークから制御する気球の製作

阿部　功*・齋藤　竜太*・武田　友美*・渡部　包文*

## A Production of Balloon Controlled by Network

Isao Abe*, Ryuta Saito*, Tomomi Takeda* and Kanefumi Watanabe*

This paper describes a production of real balloon controlled by network. By controlling a virtual balloon in a virtual space on PC, this real balloon controlled through wireless-LAN. This virtual space on PC is constructed by Java3D - Java extended API with 3D-grapchics functions. This real balloon is a helium balloon and a balloon control system under a balloon has various sensors and a control board with network interface and a wireless-LAN access point. This system is a prototype system, but will become more utility system by use case.

### 1．はじめに

近年バーチャルリアリティの分野では、急速に発展する通信技術と融合した、ネットワークバーチャルリアリティが研究されている。例えば、ネットワークを利用して、遠隔地の友人と仮想空間内で出会い会話したり、遠隔地のロボットを手元のコントローラから操作するといったことが現実に可能となってきている。また、最近ではケーブルレスの遠隔操作に適した無線LAN装置が非常に低価格で市販され、エンドユーザにも自由に利用できるようになっている。さらに、Java言語を拡張した3次元ＣＧ用APIを持ち仮想空間を構築するのに利用可能なJava3Dという技術も広まりつつある。このようにネットワークバーチャルリアリティを手軽に実現できる要素は既に存在し、今後どのようなシステムを構築するかが重要になってきている。

実際に、2001年7月に行われた通信総合研究所（CRL）の一般公開では、「ちょっと変わったインターネット」というタイトルで、「無線LANにより制御可能なラジコンカー」のデモンストレーションが行われていた。

このようなことから、本研究では無線LANを用いてネットワークから遠隔制御可能な気球を製作することを目的とする。またこのシステムを製作することで、新しい技術を取り入れたネットワークバーチャルリアリティシステムの構築技法を確立する。

### 2．方法

#### 2．1　システム概要

このシステムでは、ユーザが制御用ＰＣのキーボードで仮想空間の気球を操作することにより、Java3Dプログラム・Java通信プログラムを経由して、無線LANにより現実の気球を制御する事ができる。このJava3Dプログラムでは、現実の気球から得た情報を仮想空間に表示する。現実の気球からは、温度・湿度・高度の３つのデータを取得することができ、現実の気球の高度データにより仮想空間内の気球が移動する。これにより、Java3D仮想空間内では、現実の気球が見えなくてもその動作や状況を把握する事ができる。

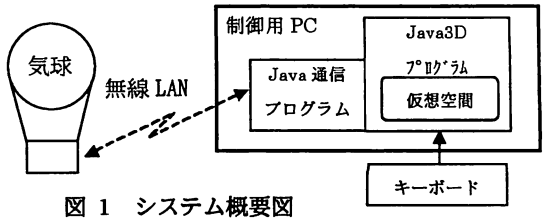



図１　システム概要図

#### 2．2　ソフトウェア構成

上記で示した機能を実現し構成が容易なように、パーソナルコンピュータ(PC)で以下のようなソフトウェアを使用し、製作した。開発ソフトウェアのJava3Dは、Sun Microsystems社が開発したJavaアプリケーション用の３次元グラフィックスAPIである。Java開発キットであるJava2 SDKは、Java3Dを利用するために必要となる。OSにWindow98を使用したが、Java3DとJava2 SDKが提供され実行可能な環境であれば、他のOSでも実現可能である。

表１　ソフトウェア構成

| | |
|---|---|
| 開発ソフトウェア | Java3D 1.2 |
| | Java2 SDK 1.3.1 |
| ＯＳ | Windows98 |

#### 2．3　ハードウェア構成

制御用ＰＣは一般に使われているDOS/V互換機を使用した。制御用ＰＣと気球間の通信には、最近ＰＣで用いられている無線LANのIEEE802.11bを使用した。

また、制御する気球は図２のような上部の気球本体と下部の気球制御部で構成されている。気球本体は、大型風船にヘリウムガスを詰めて浮かせる。気球制御部は通信部・モータ駆動部・各種センサ回路で構成され、ここにモータ４個取付ける事で、気球全体を前後左右上下に移動させることができる。さらに温度・湿度・高度（超音波）センサからの各データは、無線LANを経由して制御用PCで取得することができる。

* 山形県立産業技術短期大学校　（〒990-2473　山形市松栄２丁目２番１号）
* Yamagata College of Industry & Technology, 2-2-1,Matsuei, Yamagata-shi, Yamagata 990-2473

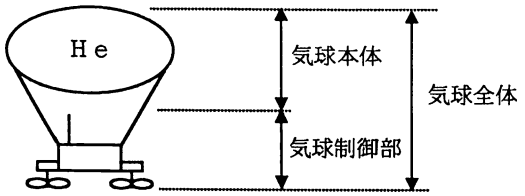



図 2　気球全体図

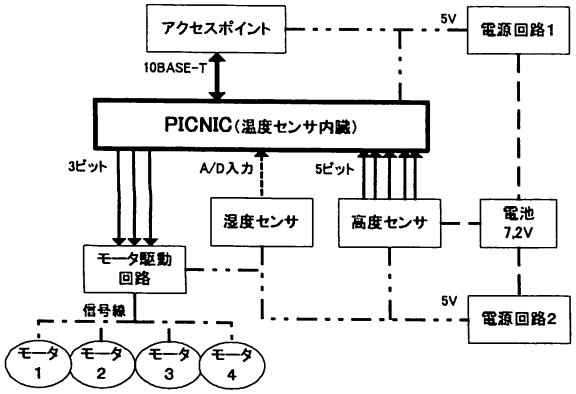



図 3　気球制御部ブロック図

## 3．実装

このシステムでは、仮想空間を構築する Java3D プログラムの作成と、現実の気球を制御するための Java 通信プログラムの作成に分けて実装した。これにより仮想空間のＧＵＩ部分と通信部分は独立性が保たれ、このようなグループでのシステム開発には非常に適している。

### 3．1　仮想空間の作成

仮想空間の物体形状とユーザインターフェースであるキーボード処理を作成するのに Java3D を利用した。

#### 3.1.1　仮想空間内のデータ処理

仮想空間内では、２秒毎に各種データを取得するスレッドと、キー操作毎にキーボード処理を行うスレッドとが、マルチスレッド処理され仮想空間内を書き換えている。

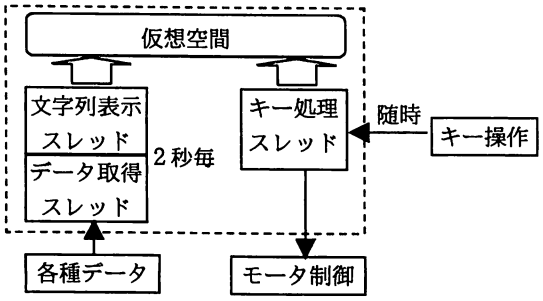



図 4　仮想空間内の仕組み

#### 3.1.2　気球の移動

仮想空間の気球と現実の気球が一緒に移動する方法を以下に示す。

(1) 上昇・下降（図５）

キーボードで上昇のキーを押すと、モータが制御され現実の気球が上昇する。それと同時に、現実の気球より高度データを取得して、そのデータをもとに仮想空間内の気球も上昇する。下降に関しても同様の方法で動作するので、現実の高度のデータを正確に反映している。

(2) 前後・左右の移動（図６）

今回の気球の構造上の問題から、気球の平面的な位置を取得することができない。そこで、例えばキーボードで前進のキーを押すと、仮想空間の気球はプログラムで前進させ、現実の気球はモータが制御され前進する。このように仮想空間内と現実の気球は見かけ上連動しているように見えるが、全く別の動きをしている。

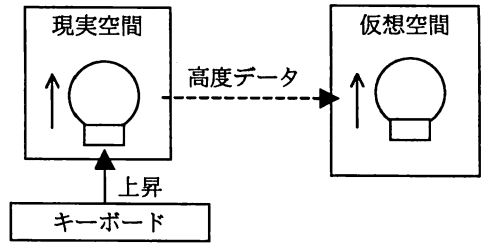



図 5　上昇・下降

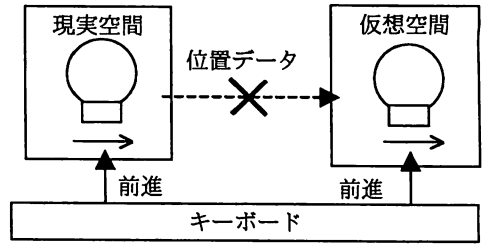



図 6　前後・左右の移動

### 3．2　通信プログラムの作成

#### 3.2.1　クラス設計

通信プログラムは無線 LAN により気球制御部の PICNIC と通信を行う。このプログラムは、実際に通信を行う PICNIC クラス、気球の動きを制御するモータ制御クラス、気球から各種データを取得するデータ取得クラスの３つで構成されている。

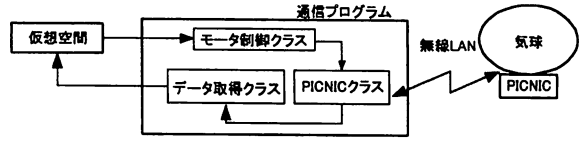



図 7　通信プログラムの概要

### 3.2.2　通信手順

このPICNICとの通信にはUDPを使用し、まずPICNICへの制御コマンドを送信した後、次にPICNICの状態を受信する手順で行われる。この時のPICNIC制御コマンドを表2、PICNICからの受信データを表3に示す。

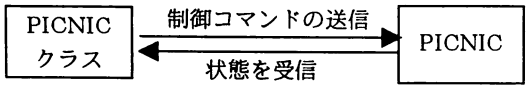



図 8　PICNICクラスの通信手順

表 2　PICNIC制御コマンド

| 目的 | | 第1バイト目 | 第2バイト目 | 第3バイト目 |
|---|---|---|---|---|
| 設定 | パラレルポートの入出力方向 | 1 ： 入力 | 0x85：ポートA | 0～7：ピン番号 |
| | | 2 ： 出力 | 0x86：ポートB | |
| | パラレルポートの値 | 1 ： HIGH | 0x05：ポートA | 0～7：ピン番号 |
| | | 2 ： LOW | 0x06：ポートB | |
| 取得 | パラレルポートの値 | 0 ： 状態取得 | | |
| | A/Dの値 | 4 ： A/Dの値取得 | 0x81 |（ピン番号*8） | 0：A/Dの蓄積時間 |

表 3　PICNIC受信データ

| バイト数 | 型 | 内容 |
|---|---|---|
| 1 | char | ポートAの状態 |
| 2 | char | ポートBの状態 |
| 3 | char | ポートAの入出力設定レジスタの状態 |
| 4 | char | ポートBの入出力設定レジスタの状態 |
| 5 | short | A/Dの値 |
| 6 | | |
| 7 | char | A/D変換用制御レジスタの状態 |
| 8 | char | 予約 |

### 3.2.3　JavaによるUDP通信

PICNICクラスはUDPにより通信を行うため、以下のような手順で行う。

(1) DatagramSocketクラスを作成する。
(2) InetAddress クラスの getByName()メソッドを使い、PICNICのIPアドレスを取得する。
(3) PICNIC制御コマンドやそのデータ長、PICNICのIPアドレスと通信ポート番号を1つのデータグラムパケットとして作成し、sendにより送信する。
(4) receiveによりPICNICからのデータを受信した後、データグラムパケットから必要なデータを取り出す。
(5) 送受信が終了したらcloseメソッドによりソケットを閉じる。

実際のプログラムは以下のようになる。

```
DatagramSocket socket = new DatagramSocket();
InetAddress address = InetAddress.getByName(ADDRESS);
DatagramPacket packet
    = new DatagramPacket( command, num, address, PORT );
socket.send( packet );                    //データ送信

DatagramPacket packetb
    = new DatagramPacket( redata, renum );
socket.receive( packetb );                //データ受信
socket.close();
```

リスト 1　javaによるUDP通信

表 4　変数の内容

| 変数名 | 内容 |
|---|---|
| ADDRESS | PICNICに設定したIPアドレス。 |
| command | PICNICの制御コマンド。 |
| num | 制御コマンドのデータ長。 |
| PORT | PICNICのパラレルポートとUDPで通信するポート番号、ここでは10000番に設定している。 |
| redata | 受信データを入れる配列。 |
| renum | 受信したデータのデータ長、8バイト |

## 3．3　気球本体の製作

気球は大型のゴム風船にヘリウムガスを充填して浮かせる事にした。実際の飛行条件1気圧・気温10℃とし、気球全体の総重量は665gのため、以下の計算式で気球の大きさを求めると、半径57cmで浮力と総重量がバランスすることが分かった。

＜計算式＞

気球の状態方程式より

空気の密度：$\rho_1 = 1.24_{(kg/m^3)}$　　・・・式①

ヘリウムの密度：$\rho_2 = 0.17_{(kg/m^3)}$　　・・・式②

気球の浮力：$F_{(kg)} = \frac{4}{3}\pi r^3 \rho_1$　　・・・式③

気球内部の重量：$W_{(kg)} = \frac{4}{3}\pi r^3 \rho_2$　　・・・式④

アルキメデスの原理より

気球の総重量：$S_{(kg)} = F - W$　　・・・式⑤

式①～⑤より半径rを求めると以下の式となる。

$$r_{(m)} = \sqrt[3]{\frac{S}{\frac{4}{3}(\rho_1 - \rho_2)}}$$

## 3．4　気球制御部の製作

気球制御部に関してはできるだけ軽量にするために、設計や部品選定でかなり制約があった。

### 3.4.1　通信部

通信部には軽量・小型なコレガ製無線LANアクセスポイントAP-11miniを使用し、既存LANに接続された制御用PCからコントロール出来るようにした。実際には軽量化のため、プラスチックケースを外して取付けている。

### 3.4.2　コントローラ部

コントローラにはLANインターフェースからI/O制御可能な物と言うことで、秋月電子通商製のPICNICを使用した。このデジタルI/Oポートを高度センサとモータ制御用に使用し、A/D入力ポートを温度・湿度センサのデータ入力用とした。（前頁 図3参照）

### 3.4.3　各種センサ

温度センサはPICNICに付属する温度センサLM35DZを使用

し、湿度センサには TDK 製 CHS-GSS を使用した。両センサとも測定データ値に対し電圧出力が線形なため非常に使いやすく、出力電圧を OP アンプにより増幅し、PICNIC の A/D 入力へ入力している。

3.4.4　高度センサ

高度センサは、超音波センサと PIC マイコンによって測距回路を製作し、それを下向きに付ける事で実現した。したがって、計測可能な高度には約 5m までという制約がある。

3.4.5　モータ駆動回路

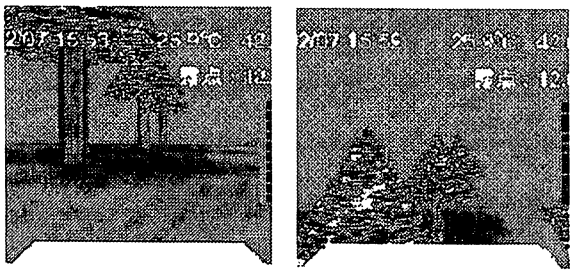
気球を上昇・下降・前進・後退・左右旋回・停止の７つの動きを実現するために、モータ駆動回路は簡単な論理回路と DC モータドライブ IC で製作した。また、モータとプロペラは、模型用の小型・軽量な物を使用した。

3.4.6　電源

電源は、無線 AP やモータ、超音波センサ回路など大電流を必要とする物が多く、なおかつ軽量小型でなければならないため、充電式の DV カメラ用リチウムイオンバッテリを利用した。

## ４．結果

制御用ＰＣのキーボードを操作すると現実の気球が上昇し、その時の高度データにより仮想空間内の気球（下部の黄色の物体）が図９のように移動することが出来た。さらに、仮想空間内には、日付・時間・温度・湿度などの様々な上方を表示することができた。



(a)上昇前　　(b)上昇後

**図 9　仮想空間内の表示**

また、図１０のように気球本体は前述の計算値とほぼ同じ大きさで浮き、飛行することができた。気球制御部も図１１に示すように、必要な機能を搭載しなおかつ軽量に実装することができた。しかし、モータ動作時に高度データが取得できないため、仮想空間内の気球と現実の気球の位置にずれが生じる事があった。原因は、モータ動作時の振動かノイズが高度センサ回路に影響を与えているようである。

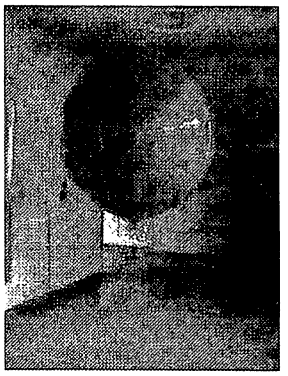


**図 10　気球の飛行状態**

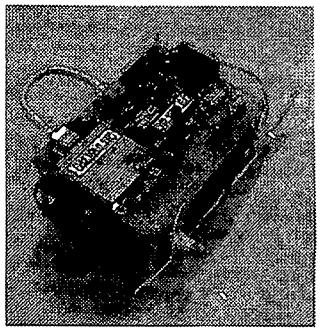


**図 11　気球制御部**

## ５．おわりに

このように無線 LAN を用いてネットワークから遠隔制御可能な気球を製作することができた。しかしこのシステムはまだ試作レベルで、気球の自己位置を認識する機能がない点や気球制御部のコントローラの機能が少ない点などまだ改善すべき点が残されている。また、このシステムの応用範囲は広く、農業、環境問題、エンタテーメントなど考えられるが、目的を絞ることでより実用的なシステムにすることが可能である。

そこで、今後は、無線通信インフラの改良・気球制御部の高機能化・ソフトウェアの改良などを行い、より現実的なシステムにしていく予定である。

**謝辞**

本研究を卒業研究とし、このシステムを作成してくれた 2001 年度情報制御システム科卒業生の齋藤竜太君、武田友美さん、渡部包文君にこの場を借りて深く感謝いたします。

**参考文献**

(1) 有路豊裕，「現実の状況を反映する仮想空間の作成」，山形県立産業技術短期大学校卒業論文，2001年3月
(2) Elliote Rusty Harold, 「JAVAネットワークプログラミング」，オライリージャパン
(3) 後閑哲也，「超音波距離計の製作」，PICマイコン活用ハンドブック，CQ出版社，pp52-60
(4)「熱気球を設計してみよう」，ニュートン，1997年4月，pp136-137
(5)「PICNIC Ver2.0 製作マニュアル」，秋月電子通商
(6)「corega Wireless LAN AP-11mini取扱説明書」，株式会社コレガ